UNIT-V-Syllabus

Bayesian Belief Network,

Concepts and mechanism,

Genetic Algorithms,

Reinforcement Learning

Active Learning

Transfer Learning,

Advance ML Applications

Bayesian Belief Network

- Bayesian belief network is key computer technology for dealing with probabilistic events and to solve a problem which has uncertainty.
- "A Bayesian network is a **probabilistic graphical model** which **represents a set of variables and their conditional dependencies** using a directed acyclic graph."
- It is also called a **Bayes network, belief network, decision network,** or **Bayesian model.**
- Bayesian networks are **probabilistic**, because these networks are built from a probability distribution, and also use **probability theory** for prediction and anomaly detection

Bayesian Belief Network

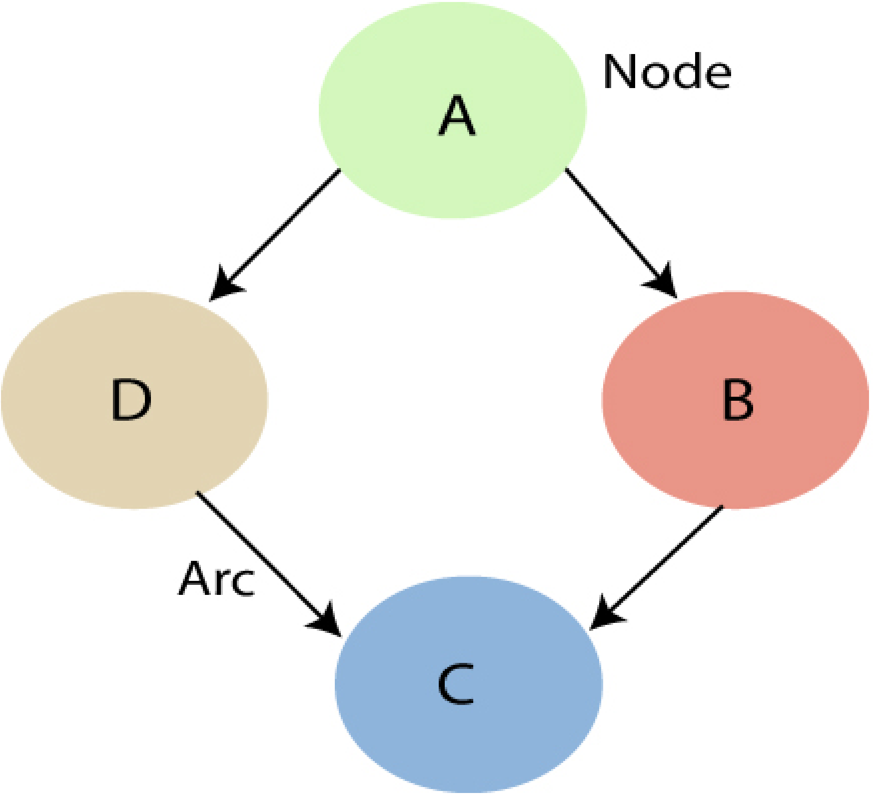
- **Real world applications are probabilistic in nature**, and **to represent** the relationship between multiple events, we need a Bayesian network.
- It can also be used in various tasks including **prediction, anomaly detection, diagnostics, automated insight, reasoning, time series prediction, and decision making** under uncertainty.

It consists of two parts:

- Directed Acyclic Graph
- Table of conditional probabilities.

Bayesian Belief Network

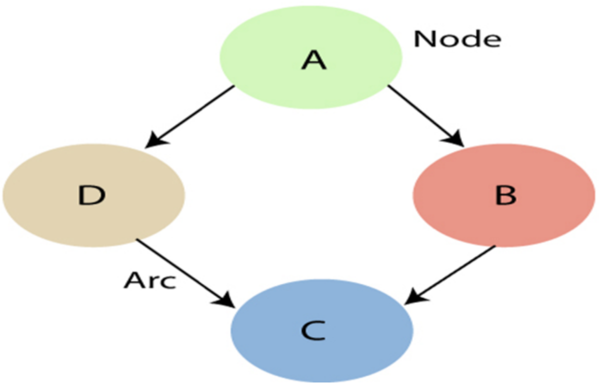
- A Bayesian network graph is made up of nodes and Arcs (directed links)



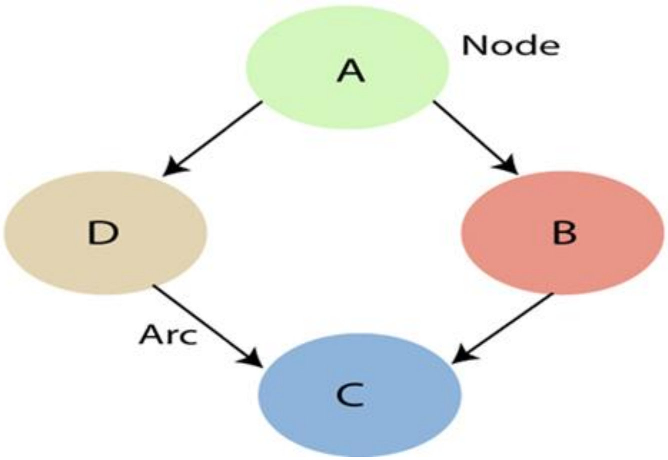
The generalized form of Bayesian network that represents **and solve decision problems under uncertain knowledge** is known as **an Influence diagram**.

Bayesian Belief Network

- **Each node** corresponds to the random variables, and **a variable** can be continuous or discrete.
- **Arc or directed arrows** represent the causal relationship or conditional probabilities between random variables. These directed links or arrows connect the pair of nodes in the graph.
- These links represent that one node directly influence the other node, and if there is no directed link that means that nodes are independent with each other

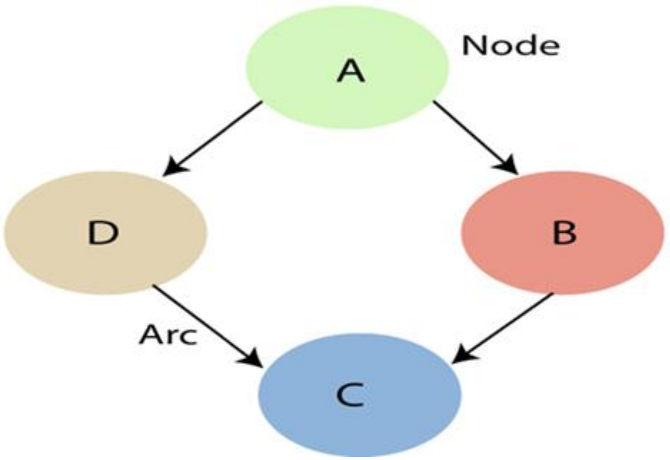


Bayesian Belief Network



- In the above diagram, A, B, C, and D are random variables represented by the nodes of the network graph.
- If we are considering node B, which is connected with node A by a directed arrow, then node **A is called the parent** of Node B.
- Node C is independent of node A.
- **The Bayesian network graph does not contain any cyclic graph. Hence, it is known as a directed acyclic graph or DAG**

Bayesian Belief Network



- In the above diagram, A, B, C, and D are random variables represented by the nodes of the network graph.
- If we are considering node B, which is connected with node A by a directed arrow, then node **A is called the parent** of Node B.
- Node C is independent of node A.
- **The Bayesian network graph does not contain any cyclic graph. Hence, it is known as a directed acyclic graph or DAG**

MIT-WPU

MIT-WPU

॥ विश्वशान्तिर्ध्रुवं ध्रुवा ॥

Bayesian network is based on Joint probability distribution and conditional probability

Probability Basics

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)} \quad (1)$$

$$P(B \mid A) = \frac{P(A \cap B)}{P(A)} \quad (2)$$

$$\boxed{P(A \mid B) = \frac{P(B \mid A)\, P(A)}{P(B)}}$$

where A and B are events and $P(B) \neq 0$.

- $P(A \mid B)$ is a conditional probability: the likelihood of event A occurring given that B is true.
- $P(B \mid A)$ is also a conditional probability: the likelihood of event B occurring given that A is true.
- $P(A)$ and $P(B)$ are the probabilities of observing A and B independently of each other; this is known as the marginal probability.

Joint Probability

- Joint probability is the likelihood of more than one event occurring at the same time P(A and B).
- The probability of event A and event B occurring together. It is the probability of the intersection of two or more events written as $p(A \cap B)$.

Example: The probability that a card is a four and red =p(four and red) = 2/52=1/26.

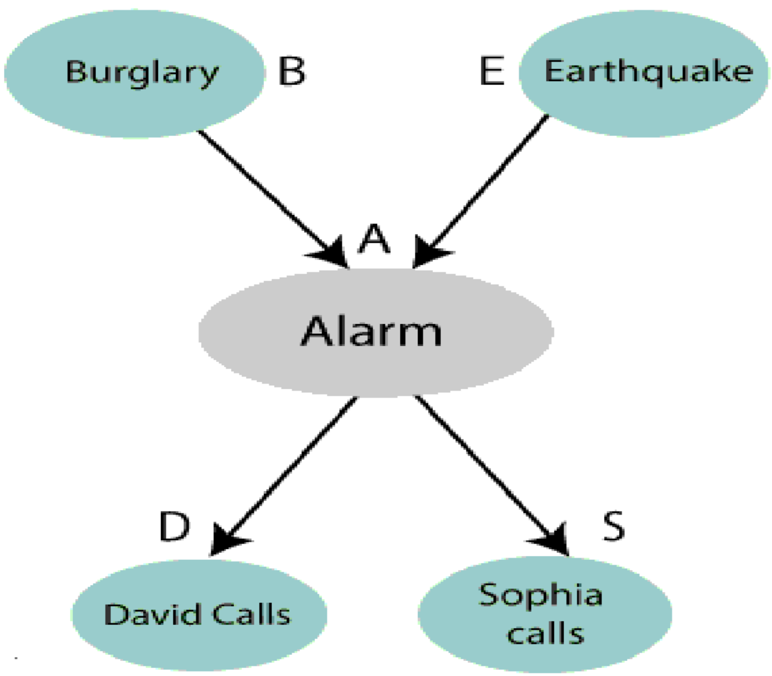
(There are two red fours in a deck of 52, the 4 of hearts and the 4 of diamonds).

Bayesian Belief Network

Example: Harry installed a new burglar alarm at his home to detect burglary.

- The alarm reliably responds at detecting a burglary but also responds for minor earthquakes.
- Harry has 2 neighbors David &Sophia, who have taken a responsibility to inform Harry at work when they hear the alarm.
- David always calls Harry when he hears the alarm, but sometimes he got confused with the phone ringing and calls at that time too.
- On the other hand, Sophia likes to listen to high music, so sometimes she misses to hear the alarm.
- Here we would like to compute the probability of Burglary Alarm.

Problem: Calculate the probability that alarm has sounded, but there is neither a burglary, nor an earthquake occurred, and David and Sophia both called the Harry.



List of all events occurring in this network:

- Burglary (B)
- Earthquake(E)
- Alarm(A)
- David Calls(D)
- Sophia calls(S)

We can write the events of problem statement in the form of probability: $P[D, S, A, B, E]$, can rewrite the above probability statement using joint probability distribution:

$$P[D, S, A, B, E]= P[D \mid S, A, B, E]. P[S, A, B, E]$$

$$=P[D \mid S, A, B, E]. P[S \mid A, B, E]. P[A, B, E]$$

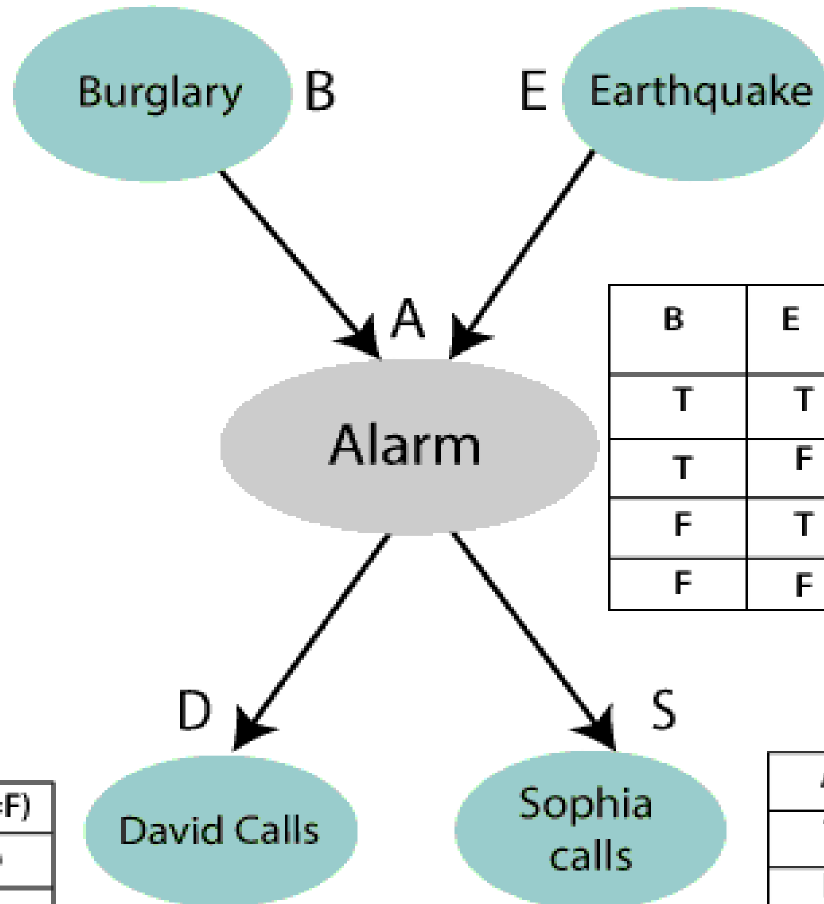
$$= P [D| A]. P [S| A, B, E]. P[A, B, E]$$

$$= P[D \mid A]. P[S \mid A]. P[A| B, E]. P[B, E]$$

$$= P[D \mid A]. P[S \mid A]. P[A| B, E]. P[B |E]. P[E]$$

T	0.002
F	0.998

T	0.001
F	0.999



B	E	P(A=T)	P(A=F)
T	T	0.94	0.06
T	F	0.95	0.04
F	T	0.69	0.69
F	F	0.999	0.999

A	P (D=T)	P (D=F)
T	0.91	0.09
F	0.05	0.95

A	P (S=T)	P (S=F)
T	0.75	0.25
F	0.02	0.98

Let's take the **observed probability for the Burglary and earthquake component:**

P(B= True) = 0.002, which is the probability of burglary.

P(B= False)= 0.998, which is the probability of no burglary.

P(E= True)= 0.001, which is the probability of a minor earthquake

P(E= False)= 0.999, Which is the probability that an earthquake not occurred.

Conditional probability table for Alarm A:

The Conditional probability of **Alarm A** depends on Burglar and earthquake:

B	E	P(A= True)	P(A= False)
True	True	0.94	0.06
True	False	0.95	0.04
False	True	0.31	0.69
False	False	0.001	0.999

Conditional probability table for David Calls:

The Conditional probability of David that he will call depends on the probability of Alarm.

A	P(D= True)	P(D= False)
True	0.91	0.09
False	0.05	0.95

Conditional probability table for Sophia Calls:

The Conditional probability of Sophia that she calls is depending on its Parent Node "Alarm."

A	P(S= True)	P(S= False)
True	0.75	0.25
False	0.02	0.98

From the formula of **joint distribution**, we can write the problem statement in the form of **probability distribution**:

$P(S, D, A, \neg B, \neg E) = P(S|A) * P(D|A) * P(A|\neg B \wedge \neg E) * P(\neg B) * P(\neg E)$.

$= 0.75 * 0.91 * 0.001 * 0.998 * 0.999$

$= 0.00068045$.

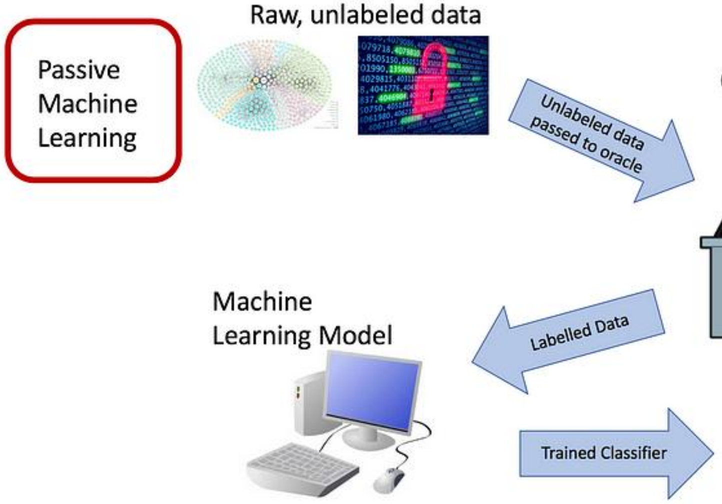
Hence, a Bayesian network can answer any query about the domain by using Joint distribution.

Active Learning

- Understanding passive/active learning
- Introduction to active learning
- Why active learning and its significance
- Active learning basic architecture/life cycle
- Active learning strategy and its working
- Use cases for active learning

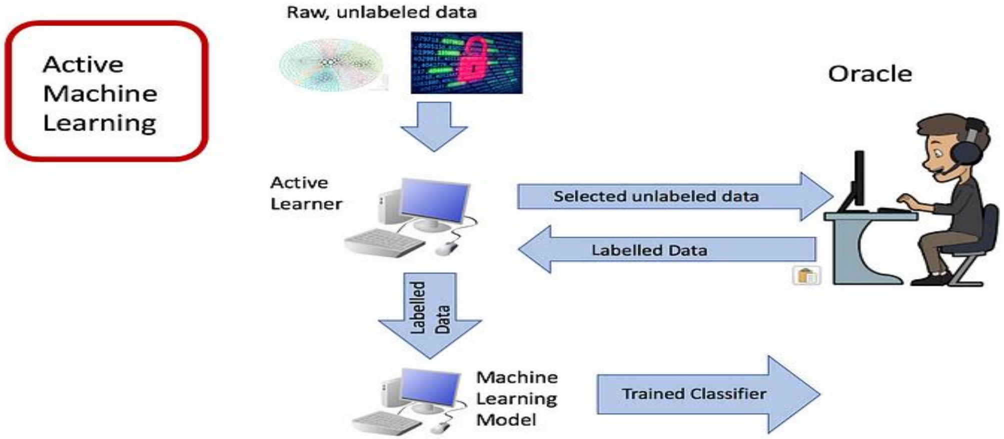
Understanding Passive Learning

- Passive learning, the standard framework in which a large quantity of labelled data is passed to the algorithm, requires significant effort in labelling the entire set of data.



Understanding Active Learning

- By using active learning, we can selectively leverage a system like crowd-sourcing, to ask human experts to selectively label some items in the data set, but not have to label the entirety.
- **The algorithm iteratively selects the most informative examples based on some value metric and sends those unlabelled examples to a labelling oracle, who returns the true labels for those queried examples back to the algorithm.**



Introduction: Active Learning

- The primary goal of machine learning is to derive general patterns from a limited amount of data.
- For most of supervised and unsupervised learning tasks, what we usually do is to gather a significant quantity of data which is randomly sampled from the underlying population distribution and then we induce a classifier or model.
- But this process is some kind of passive!
- Often the most time-consuming and costly task in the process is the gathering the data.
- Example: document classification.
- Easy to get large pool of unlabeled document, but it will take a long time for people to hand-label thousands of training document.

Introduction: Active Learning

- Now, instead of randomly picking documents to be manually labeled fro our training set, we want to choose and query documents from the pool very carefully.
- Based on this carefully choosing training data, we can improve the model’s performance very quickly.

Introduction: Active Learning

- Active learning is the subset of machine learning in which a learning algorithm can query a user interactively to label data with the desired outputs.
- A growing problem in machine learning is the large amount of unlabeled data, since data is continuously getting cheaper to collect and store.
- Active learning is the subset of machine learning in which a learning algorithm can query a user interactively to label data with the desired outputs. In active learning, the algorithm proactively selects the subset of examples to be labeled next from the pool of unlabeled data.

Introduction: Active Learning

- The fundamental belief behind the active learner algorithm concept is that an ML algorithm could potentially reach a higher level of accuracy while using a smaller number of training labels if it were allowed to choose the data it wants to learn from.
- Therefore, active learners are allowed to interactively pose queries during the training stage.
- These queries are usually in the form of unlabeled data instances and the request is to a human annotator to label the instance.
- This makes active learning part of the human-in-the-loop paradigm, where it is one of the most powerful examples of success.

Why Active Learning?

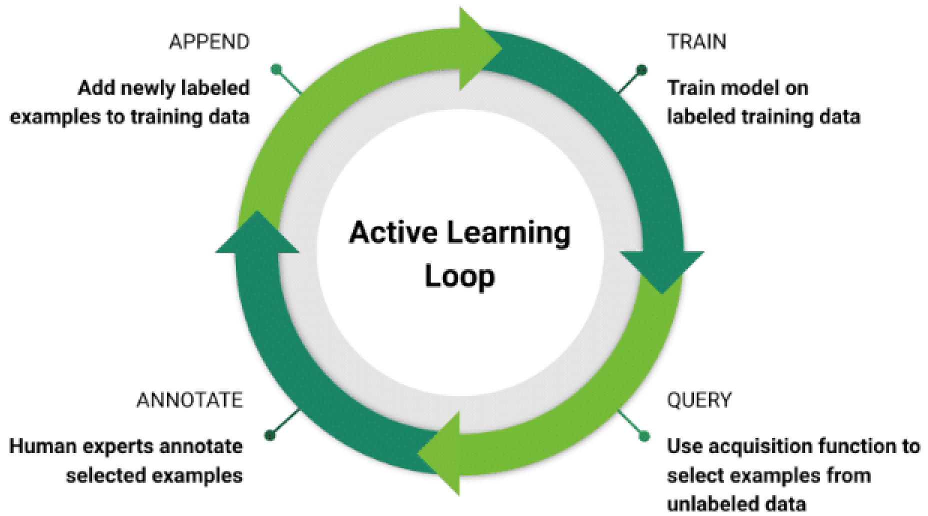
- Most supervised machine learning models require large amounts of data to be trained with good results. And even if this statement sounds naive, most companies struggle to provide their data scientists this data, in particular labelled data. The latter is key to train any supervised model and can become the main bottleneck for any data team.

- In most cases, data scientists are provided with a big, unlabelled data sets and are asked to train well-performing models with them. Generally, the amount of data is too large to manually label it, and it becomes quite challenging for data teams to train good supervised models with that data.

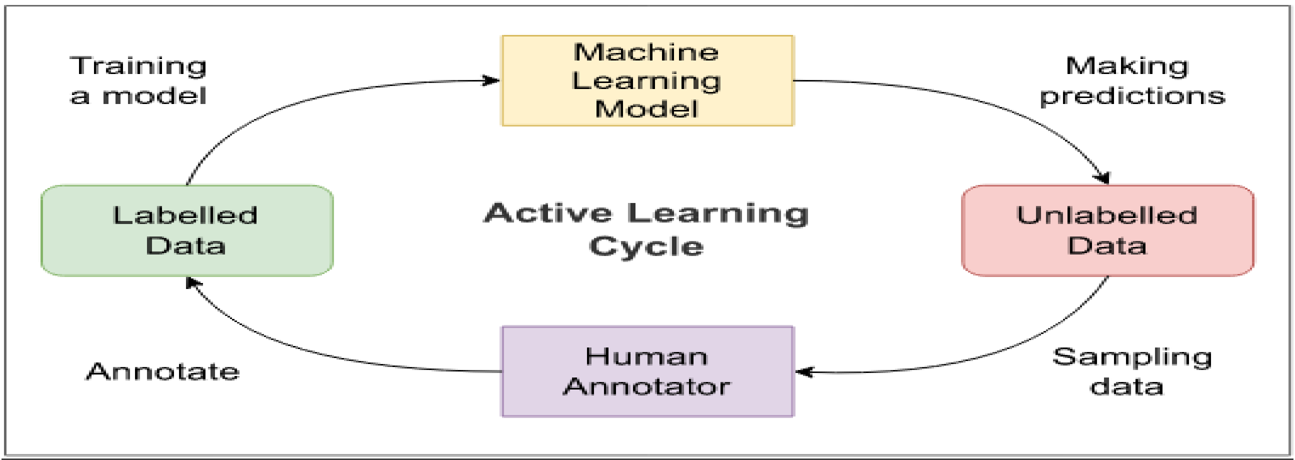
Significance of Active Learning

- *Active Learning is a “human-in-the-loop” type of Deep Learning framework that uses a large dataset of which only a small portion (say 10%) is labeled for model training. Say there is a dataset of 1,000 samples, of which 100 are labeled. An Active Learning-based model will train on the 100 samples and make predictions on the rest of the 900 samples (test set). Suppose, of these 900 samples, the confidence in prediction was very low for 10 samples. The model will now ask a human user to provide it with the labels for these 10 samples. That is, an Active Learning framework is interactive, and that’s how the name “Active” was coined.*

Active Learning: Basic Architecture



Active Learning Cycle



Because an active learning approach starts with a small labeled dataset, the initial predictions that the model makes on the unlabeled data won't be very. However, this iterative training, testing, identifying uncertainty, annotating, and retraining feedback loop continues until the model reaches an acceptable performance threshold. At that point, the model's predictions with a high level of certainty can be sent downstream for use in production while the others are sent back to the annotators, keeping the loop active and constantly improving.

Active Learning: Motivation

- Active learning is the name used for the process of prioritising the data which needs to be labelled in order to have the highest impact to training a supervised model.
- Active learning can be used in situations where the amount of data is too large to be labelled and some priority needs to be made to label the data in a smart way.
- Active learning is closer to traditional supervised learning, a type of semi-supervised learning, meaning models are trained using both labeled and unlabeled data.
- The idea behind semi-supervised learning is that labeling just a small sample of data might result in the same accuracy or better than fully labeled training data.
- The only challenge is determining what that sample is: Active learning machine learning is all about labeling data dynamically and incrementally during the training phase so that the algorithm can identify what label would be the most beneficial for it to learn from.

Active Learning Strategy

- Steps for active learning
  - There are multiple approaches studied in the literature on how to prioritise data points when labelling and how to iterate over the approach.
  - We will nevertheless only present the most common and straightforward methods.
  - The steps to use active learning on an unlabelled data set are:
  - The first thing which needs to happen is that a very small subsample of this data needs to be manually labelled.
  - Once there is a small amount of labelled data, the model needs to be trained on it.
  - The model is of course not going to be great but will help us get some insight on which areas of the parameter space need to be labelled first to improve it.

Active Learning Strategy

- After the model is trained, the model is used to predict the class of each remaining unlabelled data point.
- A score is chosen on each unlabelled data point based on the prediction of the model.
- Once the best approach has been chosen to prioritise the labelling, this process can be iteratively repeated: a new model can be trained on a new labelled data set, which has been labelled based on the priority score.
- Once the new model has been trained on the subset of data, the unlabelled data points can be ran through the model to update the prioritisation scores to continue labelling.
- In this way, one can keep optimising the labelling strategy as the models become better and better.

How does Active Learning work?

- Active learning works in a few different situations, basically, the decision of whether or not to query each specific label depends on whether the gain from querying the label is greater than the cost of obtaining that information.
- This decision making, in practice, can take a few different forms based on the data scientist's budget limit and other factors.
- The three categories of active learning are:
  - **Stream based selective sampling**
  - **Pool-based sampling**
  - **Membership query synthesis**

How does Active Learning work?

- **Stream-based selective sampling**
- In this scenario, the algorithm determines if it would be beneficial enough to query for the label of a specific unlabeled entry in the dataset.
- While the model is being trained, it is presented with a data instance and immediately decides if it wants to query the label.
- This approach has a natural disadvantage that comes from the lack of guarantee that the data scientist will stay within budget.

How does Active Learning work?

- **Pool-based sampling**
- This is the most well known scenario for active learning.
- In this sampling method, the algorithm attempts to evaluate the entire dataset before it selects the best query or set of queries.
- The active learner algorithm is often initially trained on a fully labeled part of the data which is then used to determine which instances would be most beneficial to insert into the training set for the next active learning loop.
- The downside of this method is the amount of memory it can require.

How does Active Learning work?

- **Membership query synthesis**
- This scenario is not applicable to all cases, because it involves the generation of synthetic data.
- The active learner in this method is allowed to create its own examples for labeling.
- This method is compatible with problems where it is easy to generate a data instance.

Active Learning Use Cases

- Active learning has found a number of applications in areas such as text categorization, document classification, and image recognition. It has also been used for cancer detection and drug discovery.
- **Text Categorization**
- One of the most common applications of active learning is text categorization, which is the task of assigning a category to a piece of text. In this application, the categories are usually a set of predefined labels such as “news”, “sports”, “entertainment”, and “opinion”. The goal is to automatically assign each piece of text to one of these categories.
- **Document Classification**
- Active learning can also be used for document classification, which is the task of automatically assigning a class to a document. In this application, the classes are usually a set of predefined labels such as “technical document”, “marketing document”, and “legal document”.

Active Learning Use Cases

- **Image Recognition**
- Image recognition is another area where active learning can be used. In this example, we have an image and we'd like our annotators to label only relevant regions in the image. In other words, we need to make sure that each labeled region contributes maximum information for classifying the image. To achieve this objective, active learning will pick up the most interesting regions from unlabelled data and let them be processed by annotators.
- This way, annotators don't waste any time on labeling redundant parts of an image that would have remained untagged if they were just blindly assigning labels to all regions in an image.

Reinforcement learning

- Reinforcement learning is an area of Machine Learning.
- It is about taking suitable action to maximize reward in a particular situation.
- It is employed by various software and machines to find the best possible behavior or path it should take in a specific situation.
- Reinforcement learning differs from supervised learning in a way that in supervised learning the training data has the answer key with it so the model is trained with the correct answer itself whereas in reinforcement learning, there is no answer but the reinforcement agent decides what to do to perform the given task.
- In the absence of a training dataset, it is bound to learn from its experience.

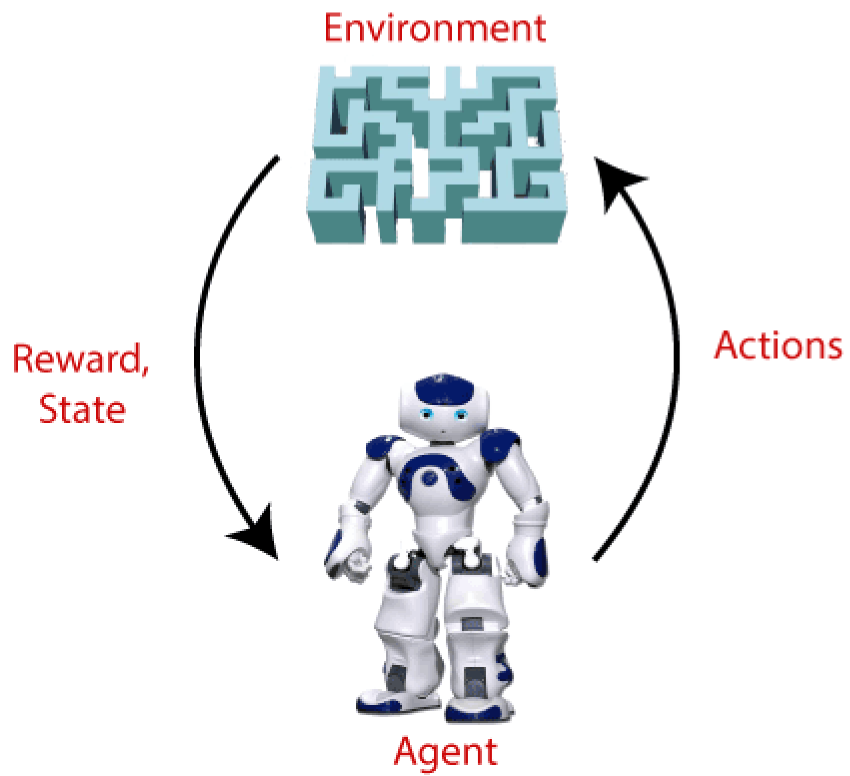
•Reinforcement Learning (RL) is the science of decision making.

•It is about learning the optimal behavior in an environment to obtain maximum reward.

•In RL, the data is accumulated from machine learning systems that use a trial-and-error method.

•Data is not part of the input that we would find in supervised or unsupervised machine learning.

- Reinforcement Learning is a feedback-based Machine learning technique in which an agent learns to behave in an environment by performing the actions and seeing the results of actions. For each good action, the agent gets positive feedback, and for each bad action, the agent gets negative feedback or penalty.
- In Reinforcement Learning, the agent learns automatically using feedbacks without any labeled data, unlike supervised learning.
- Since there is no labeled data, so the agent is bound to learn by its experience only.


Environment
Reward,
State
Actions
Agent

- Reinforcement learning uses algorithms that learn from outcomes and decide which action to take next.
- After each action, the algorithm receives feedback that helps it determine whether the choice it made was correct, neutral or incorrect.
- It is a good technique to use for automated systems that have to make a lot of small decisions without human guidance.
- Reinforcement learning is an autonomous, self- teaching system that essentially learns by trial and error.
- It performs actions with the aim of maximizing rewards, or in other words, it is learning by doing in order to achieve the best outcomes.

Elements of Reinforcement Learning

1. Policy
2. Reward function
3. Value function
4. Model of the environment

Policy: Policy defines the learning agent behavior for given time period. It is a mapping from perceived states of the environment to actions to be taken when in those states.

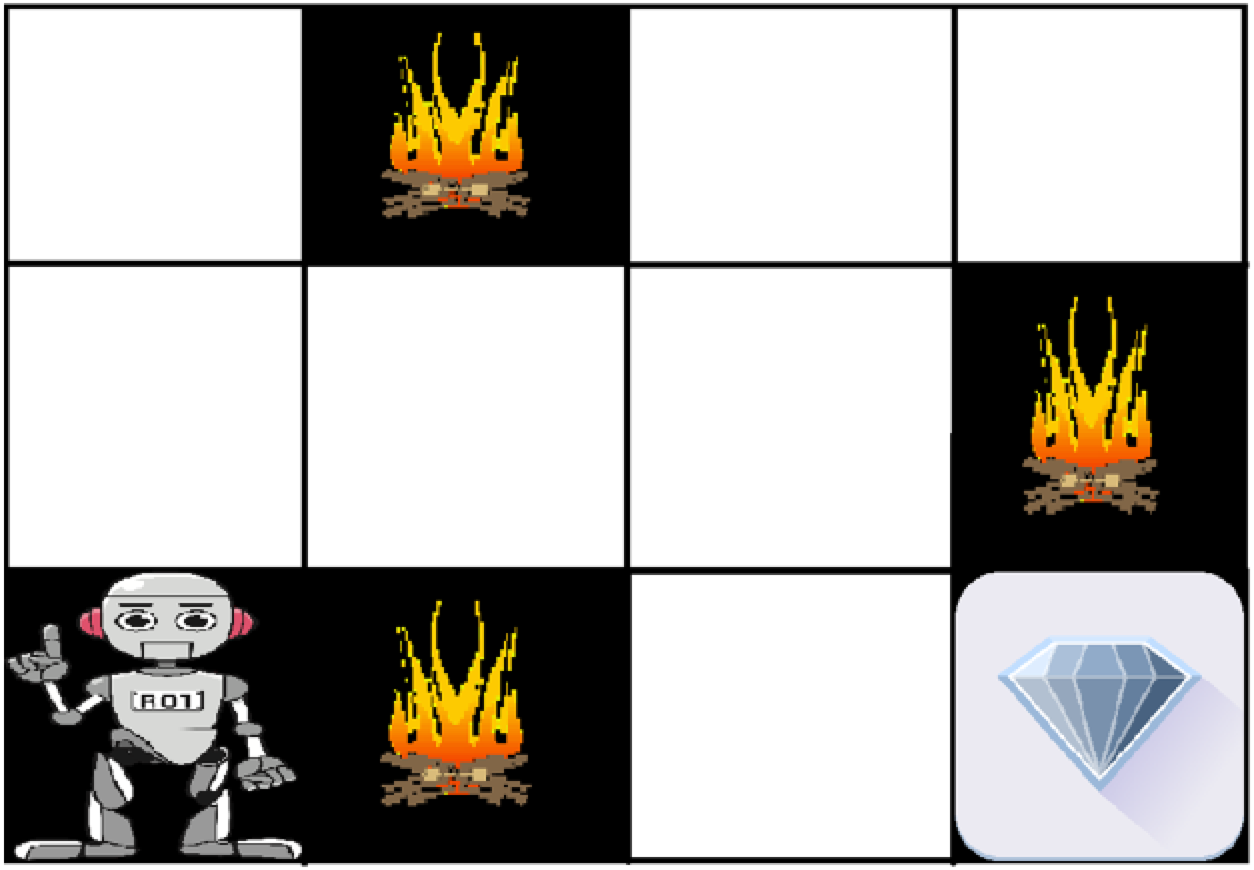
Reward function: Reward function is used to define a goal in a reinforcement learning problem.A reward function is a function that provides a numerical score based on the state of the environment

Value function: Value functions specify what is good in the long run. The value of a state is the total amount of reward an agent can expect to accumulate over the future, starting from that state.

Model of the environment: Models are used for planning.

The reinforcement learning problem model is an agent continuously interacting with an environment. The agent and the environment interact in a sequence of time steps. At each time step t, the agent receives the state of the environment and a scalar numerical reward for the previous action, and then the agent then selects an action.

•**Example:** The problem is as follows: We have an agent and a reward, with many hurdles in between. The agent is supposed to find the best possible path to reach the reward. The following problem explains the problem more easily.



- The image shows the robot, diamond, and fire.
- The goal of the robot is to get the reward that is the diamond and avoid the hurdles that are fired.
- The robot learns by trying all the possible paths and then choosing the path which gives him the reward with the least hurdles.
- Each right step will give the robot a reward and each wrong step will subtract the reward of the robot.
- The total reward will be calculated when it reaches the final reward that is the diamond.

Main points in Reinforcement Learning

- Input: The input should be an initial state from which the model will start
- Output: There are many possible outputs as there are a variety of solutions to a particular problem
- Training: The training is based upon the input, The model will return a state and the user will decide to reward or punish the model based on its output.
- The model keeps continues to learn.
- The best solution is decided based on the maximum reward.

MIT-WPU

।। विश्वशान्तिर्ध्रुवं ध्रुवा ।।

Reinforcement learning	Supervised learning
Reinforcement learning is all about making decisions sequentially. In simple words, we can say that the output depends on the state of the current input and the next input depends on the output of the previous input	In Supervised learning, the decision is made on the initial input or the input given at the start
In Reinforcement learning decision is dependent, So we give labels to sequences of dependent decisions	In supervised learning the decisions are independent of each other so labels are given to each decision.
Example: Chess game,text summarization	Example: Object recognition,spam detetction

Application of Reinforcement Learnings

1. Robotics: Robots with pre-programmed behavior are useful in structured environments, such as the assembly line of an automobile manufacturing plant, where the task is repetitive in nature.

2. A master chess player makes a move. The choice is informed both by planning, anticipating possible replies and counter replies.

3. An adaptive controller adjusts parameters of a petroleum refinery's operation in real time.

Advantages of Reinforcement learning

1. Reinforcement learning can be used to solve very complex problems that cannot be solved by conventional techniques.
2. The model can correct the errors that occurred during the training process.
3. In RL, training data is obtained via the direct interaction of the agent with the environment
4. Reinforcement learning can handle environments that are non-deterministic, meaning that the outcomes of actions are not always predictable. This is useful in real-world applications where the environment may change over time or is uncertain.
5. Reinforcement learning can be used to solve a wide range of problems, including those that involve decision making, control, and optimization.
6. Reinforcement learning is a flexible approach that can be combined with other machine learning techniques, such as deep learning, to improve performance.

Disadvantages of Reinforcement learning

1. Reinforcement learning is not preferable to use for solving simple problems.

2. Reinforcement learning needs a lot of data and a lot of computation

3. Reinforcement learning is highly dependent on the quality of the reward function. If the reward function is poorly designed, the agent may not learn the desired behavior.

4. Reinforcement learning can be difficult to debug and interpret. It is not always clear why the agent is behaving in a certain way, which can make it difficult to diagnose and fix problems.

Genetic algorithms

- As early as 1962, John Holland's work on adaptive systems laid the foundation for later developments.
- By the 1975, the publication of the book Adaptation in Natural and Artificial Systems, by Holland and his students and colleagues

- early to mid-1980s, genetic algorithms were being applied to a broad range of subjects.
- In 1992 John Koza has used genetic algorithm to evolve programs to perform certain tasks. He called his method "genetic programming" (GP).

- A genetic algorithm (or GA) is a search technique used in computing to find true or approximate solutions to optimization and search problems.
- (GA)s are categorized as global search heuristics.
- (GA)s are a particular class of evolutionary algorithms that use techniques inspired by evolutionary biology such as inheritance, mutation, selection, and crossover (also called recombination)

- The evolution usually starts from a population of randomly generated individuals and happens in generations. •

- In each generation, the fitness of every individual in the population is evaluated, multiple individuals are selected from the current population (based on their fitness), and modified to form a new population

- The new population is used in the next iteration of the algorithm. •
- The algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population.

- Individual - Any possible solution •
- Population - Group of all individuals •
- Fitness – Target function that we are optimizing (each individual has a fitness) •
- Trait - Possible aspect (features) of an individual • Genome - Collection of all chromosomes (traits) for an individual

- Start with a large “population” of randomly generated “attempted solutions” to a problem •
- Repeatedly do the following: – Evaluate each of the attempted solutions – (probabilistically) keep a subset of the best solutions – Use these solutions to generate a new population •
- Quit when you have a satisfactory solution (or you run out of time

- Suppose we want to maximize the number of ones in a string of l binary digits
- It may seem so because we know the answer in advance
- However, we can think of it as maximizing the number of correct answers, each encoded by 1, to l yes/no difficult questions

- An individual is encoded (naturally) as a string of l binary digits •
- The fitness f of a candidate solution to the MAXONE problem is the number of ones in its genetic code •
- We start with a population of n random strings. Suppose that l = 10 and n = 6

Example (initialization)

We toss a fair coin 60 times and get the following initial population:

$$s_1 = 1111010101 \quad f(s_1) = 7$$

$$s_2 = 0111000101 \quad f(s_2) = 5$$

$$s_3 = 1110110101 \quad f(s_3) = 7$$

$$s_4 = 0100010011 \quad f(s_4) = 4$$

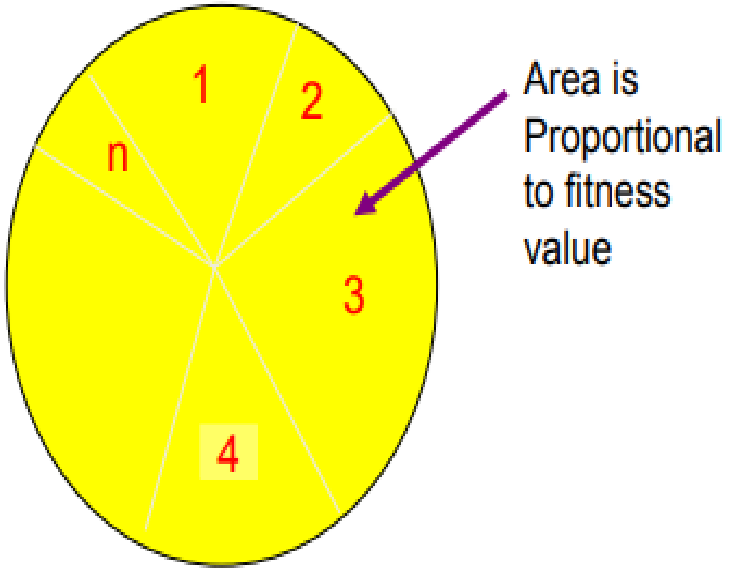
$$s_5 = 1110111101 \quad f(s_5) = 8$$

$$s_6 = 0100110000 \quad f(s_6) = 3$$

Step 1: Selection

We randomly (using a biased coin) select a subset of the individuals based on their fitness:

Individual i will have a probability to be chosen $\dfrac{f(i)}{\sum_i f(i)}$



Selected set

Suppose that, after performing selection, we get the following population:

$s_1` = 1111010101\ (s_1)$

$s_2` = 1110110101\ (s_3)$

$s_3` = 1110111101\ (s_5)$

$s_4` = 0111000101\ (s_2)$

$s_5` = 0100010011\ (s_4)$

$s_6` = 1110111101\ (s_5)$

- Next we mate strings for crossover. For each couple we first decide (using some pre-defined probability, for instance 0.6) whether to actually perform the crossover or not •
- If we decide to actually perform crossover, we randomly extract the crossover points, for instance 2 and 5

Crossover result

Before crossover:

$s_1` = 1111010101$ $s_2` = 1110110101$

After crossover:

$s_1`` = 1110110101$ $s_2`` = 1111010101$

Step 3: mutations

The final step is to apply random mutations: for each bit that we are to copy to the new population we allow a small probability of error (for instance 0.1)

Initial strings	After mutating
s_1`` = 1110110101	s_1``` = 1110100101
s_2`` = 1111010101	s_2``` = 1111110100
s_3`` = 1110111101	s_3``` = 1110101111
s_4`` = 0111000101	s_4``` = 0111000101
s_5`` = 0100011101	s_5``` = 0100011101
s_6`` = 1110110011	s_6``` = 1110110001

- In one generation, the total population fitness changed from 34 to 37, thus improved by ~9%
- At this point, we go through the same process all over again, until a stopping criterion is met

- Methods of representation •
- Methods of selection •
- Methods of Reproduction

Common representation methods

- Binary strings. •
- Arrays of integers (usually bound) •
- Arrays of letters

Methods of Selection

There are many different strategies to select the individuals to be copied over into the next generation

Methods of Selection

Roulette-wheel selection. •

Elitist selection. •

Fitness-proportionate selection. •

Scaling selection. •

Rank selection.

Roulette wheel selection

-Conceptually, this can be represented as a game of roulette - each individual gets a slice of the wheel, but more fit ones get larger slices than less fit ones.

Methods of Reproduction

There are primary methods:

–Crossover

–Mutation

Methods of Reproduction: Crossover

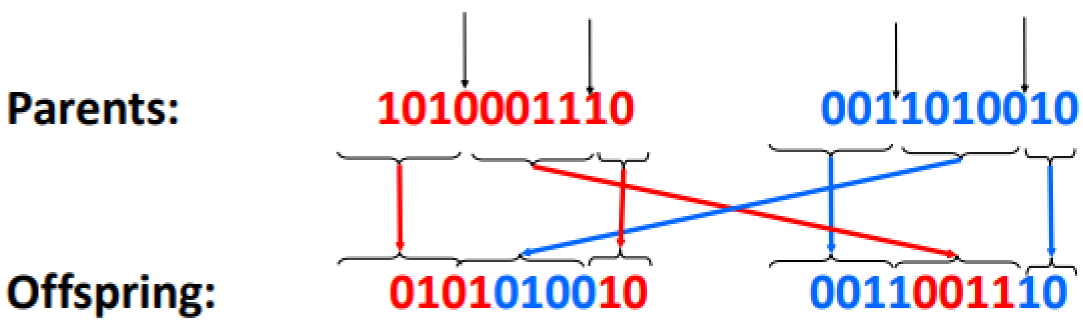
Two parents produce two offspring –

Two options:

1. The chromosomes of the two parents are copied to the next generation
2. The two parents are randomly recombined (crossed-over) to form new offsprings

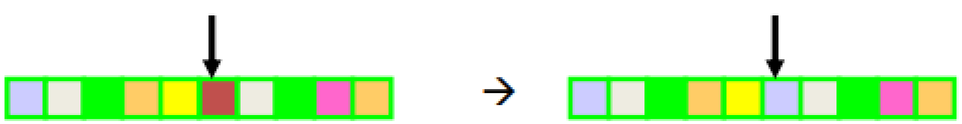
Two-point crossover

- Avoids cases where genes at the beginning and end of a chromosome are always split



Methods of Reproduction: Mutations

– Generating new offspring from single parent



- Concept is easy to understand •
- Modular, separate from application •
- Supports multi-objective optimization •
- Always an answer; answer gets better with time. •
- Easy to exploit previous or alternate solutions •
- Flexible building blocks for hybrid applications.

GA Applications

Domain	Application Type
Control	Gas pipeline, missile evasion
Design	Aircraft design, keyboard configuration, communication networks
Game playing	Poker, checkers
Security	Encryption and Decryption
Robotics	Trajectory planning

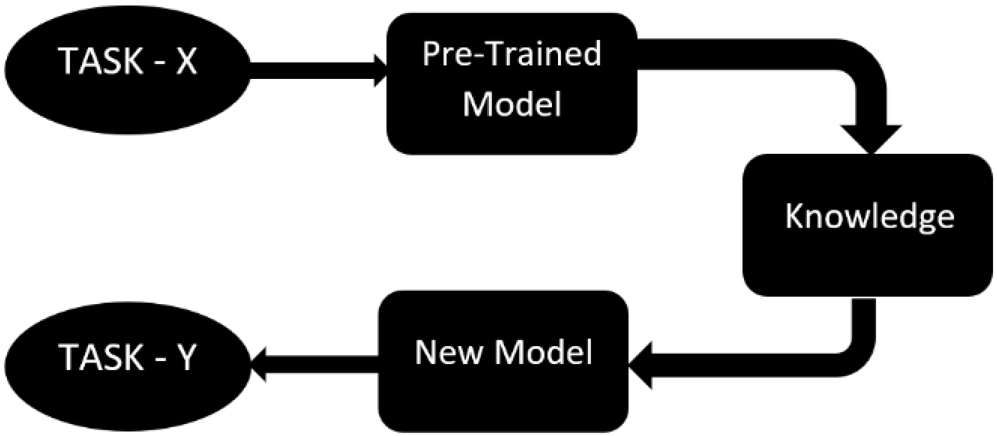
Introduction to Transfer Learning in ML

- Humans are extremely skilled at transferring knowledge from one task to another.
- This means that when we face a new problem or task, we immediately recognize it and use the relevant knowledge we have gained from previous learning experiences.
- This makes it easy to complete our tasks quickly and efficiently.
  - If a user can ride a bike and are asked to drive a motorbike, this is a good example. Their experience with riding a bike will be helpful in such situations. They can balance the bike and steer the motorbike.
  - This will make it easier than if they were a complete beginner. These lessons are extremely useful in real life because they make us better and allow us to gain more experience.
- The same approach was used to introduce **Transfer learning** into machine learning. This involves using knowledge from a task to solve a problem in the target task.
- Although most machine learning algorithms are designed for a single task, there is an ongoing interest in developing transfer learning algorithms.

Why Transfer Learning?

- One curious feature that many deep neural networks built on images share is the ability to detect edges, colours, intensities variations, and other features in the early layers.
- These features are not specific to any particular task or dataset.
- It doesn't matter what kind of image we are using to detect lions or cars.
- These low-level features must be detected in both cases. These features are present regardless of whether the image data or cost function is exact.
- These features can be learned in one task, such as detecting lions. They can also be used to detect humans. Transfer learning is exactly what this is.

Block Diagram



Freezed and Trainable Layers:

- The freezing of layers characterizes transfer learning. When a layer is unavailable to train, it is called a "Frozen Layer". It can be either a CNN layer or a hidden layer. Layers that have not been frozen are subject to regular training. Training will not update the layers' weights that have been frozen.

Let's look at all scenarios where the target task size and data set differ from the base network.

- **The target dataset is smaller than the base network data:** Because the target dataset is so small, we can fine-tune our pre-trained network using this target dataset. This could lead to overfitting. There may also be changes in the number of classes for the target task. In such cases, we may need to remove some layers that are not fully connected from the end and add a new layer that is fully connected. We now freeze the rest of our model and train only newly added layers.
- **The target dataset is large, similar to the base training dataset:** If the dataset is large enough to hold a pre-trained model, there won't be any chance of overfitting. This is where the last fully connected layer is removed, and a new fully connected layer with the correct number of classes is added. The entire model is now trained on a new dataset. This allows the model to be tuned on a large new dataset while keeping the architecture unchanged.

- **The target dataset is smaller than the base network data, and therefore, it is different:** The target dataset is unique, so pre-trained models with high-level features will not work. We can remove the most layers from the end of a pre-trained model and add layers that satisfy the number of classes in the new dataset. We can then use the low-level features of the pre-trained model to train the remaining layers to adapt to a new dataset. Sometimes it can be beneficial to train the entire network, even after adding a layer at the end.
- **The target dataset is larger than the base network data:** As the target network is complex and diverse, it is best to remove layers from pre-trained networks and add layers that satisfy a number of classes. Then train the entire network without freezing any layers.

Examples of transfer learning for machine learning

- Although an emerging technique, transfer learning is already being utilised in a range of fields within machine learning. Whether strengthening natural language processing or computer vision, transfer learning already has a range of real-world usage.
- Examples of the areas of machine learning that utilise transfer learning include:
  - Natural language processing
  - Computer vision
  - Neural networks

Transfer learning in natural language processing

- Natural language processing is the ability of a system to understand and analyze human language, whether through audio or text files. It's an important part of improving how humans and systems interact. Natural language processing is intrinsic to everyday services like voice assistants, speech recognition software, automated captions, translations, and language contextualization tools.
- Transfer learning is used in a range of ways to strengthen machine learning models that deal with natural language processing. Examples include simultaneously training a model to detect different elements of language, or embedding pre-trained layers which understand specific dialects or vocabulary.
- Transfer learning can also be used to adapt models across different languages. Aspects of models trained and refined based on the English language can be adapted for similar languages or tasks. Digitized English language resources are very common, so models can be trained on a large dataset before elements are transferred to a model for a new language.

Transfer learning in computer vision

- Computer vision is the ability of systems to understand and take meaning from visual formats such as videos or images. Machine learning algorithms are trained on huge collections of images to be able to recognise and categorise image subjects. Transfer learning in this case will take the reusable aspects of a computer vision algorithm and apply it to a new model.
- Transfer learning can take the accurate models produced from large training datasets and help apply it to smaller sets of images. This includes transferring the more general aspects of the model, such as the process for identifying the edges of objects in images. The more specific layer of the model which deals with identifying types of objects or shapes can then be trained. The model's parameters will need to be refined and optimised, but the core functionality of the model will have been set through transfer learning.

Transfer learning in neural networks

- Artificial neural networks are an important aspect of deep learning, an area of machine learning attempting to simulate and replicate the functions of the human brain. The training of neural networks takes a huge amount of resources because of the complexity of the models. Transfer learning is used to make the process more efficient and lower the resource demand.
- Any transferable knowledge or features can be moved between networks to streamline the development of new models. The application of knowledge across different tasks or environments is an important part of building such a network. Transferred learning will usually be limited to general processes or tasks which stay viable in different environments.

Advantages of Transfer Learning :

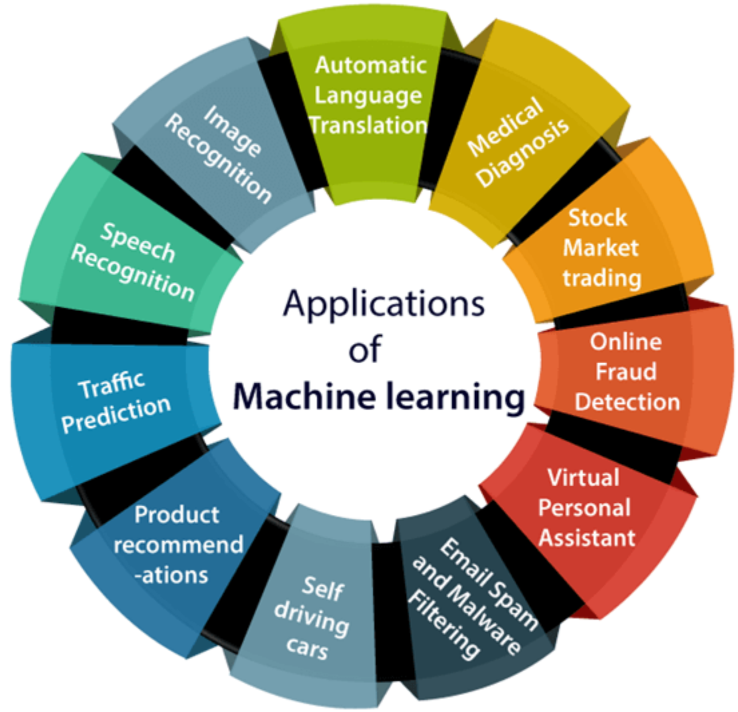
- **Speed up the training process:** By using a pre-trained model, the model can learn more quickly and effectively on the second task, as it already has a good understanding of the features and patterns in the data.
- **Better performance**: Transfer learning can lead to better performance on the second task, as the model can leverage the knowledge it has gained from the first task.
- **Handling small datasets:** When there is limited data available for the second task, transfer learning can help to prevent overfitting, as the model will have already learned general features that are likely to be useful in the second task.

Disadvantages:

- **Domain mismatch:** The pre-trained model may not be well-suited to the second task if the two tasks are vastly different or the data distribution between the two tasks is very different.
- **Overfitting**: Transfer learning can lead to overfitting if the model is fine-tuned too much on the second task, as it may learn task-specific features that do not generalize well to new data.
- **Complexity**: The pre-trained model and the fine-tuning process can be computationally expensive and may require specialized hardware.

Applications of Machine learning

- We are using machine learning in our daily life even without knowing it such as Google Maps, Google assistant, Alexa, etc. Below are some most trending real-world applications of Machine Learning.



1. Image Recognition:

- Image recognition is one of the most common applications of machine learning. It is used to identify objects, persons, places, digital images, etc. The popular use case of image recognition and face detection is, **Automatic friend tagging suggestion**:
- Facebook provides us a feature of auto friend tagging suggestion. Whenever we upload a photo with our Facebook friends, then we automatically get a tagging suggestion with name, and the technology behind this is machine learning's **face detection** and **recognition algorithm**.
- It is based on the Facebook project named "**Deep Face**," which is responsible for face recognition and person identification in the picture.

2. Speech Recognition

- While using Google, we get an option of "**Search by voice**," it comes under speech recognition, and it's a popular application of machine learning.
- Speech recognition is a process of converting voice instructions into text, and it is also known as "**Speech to text**", or "**Computer speech recognition**." At present, machine learning algorithms are widely used by various applications of speech recognition. **Google assistant**, **Siri**, **Cortana**, and **Alexa** are using speech recognition technology to follow the voice instructions.

3. Traffic prediction:

- If we want to visit a new place, we take help of Google Maps, which shows us the correct path with the shortest route and predicts the traffic conditions.
- It predicts the traffic conditions such as whether traffic is cleared, slow-moving, or heavily congested with the help of two ways:
  - **Real Time location** of the vehicle form Google Map app and sensors
  - **Average time has taken** on past days at the same time.
- Everyone who is using Google Map is helping this app to make it better. It takes information from the user and sends back to its database to improve the performance.

4. Product recommendations:

- Machine learning is widely used by various e-commerce and entertainment companies such as **Amazon**, **Netflix**, etc., for product recommendation to the user. Whenever we search for some product on Amazon, then we started getting an advertisement for the same product while internet surfing on the same browser and this is because of machine learning.
- Google understands the user interest using various machine learning algorithms and suggests the product as per customer interest.
- As similar, when we use Netflix, we find some recommendations for entertainment series, movies, etc., and this is also done with the help of machine learning.

5. Self-driving cars:

- One of the most exciting applications of machine learning is self-driving cars. Machine learning plays a significant role in self-driving cars. Tesla, the most popular car manufacturing company is working on self-driving car. It is using unsupervised learning method to train the car models to detect people and objects while driving.

6. Virtual Personal Assistant:

- We have various virtual personal assistants such as **Google assistant**, **Alexa**, **Cortana**, **Siri**. As the name suggests, they help us in finding the information using our voice instruction. These assistants can help us in various ways just by our voice instructions such as Play music, call someone, Open an email, Scheduling an appointment, etc.
- These virtual assistants use machine learning algorithms as an important part.
- These assistant record our voice instructions, send it over the server on a cloud, and decode it using ML algorithms and act accordingly.

7. Medical Diagnosis:

- In medical science, machine learning is used for diseases diagnoses. With this, medical technology is growing very fast and able to build 3D models that can predict the exact position of lesions in the brain.
- It helps in finding brain tumors and other brain-related diseases easily.

8. Automatic Language Translation:

- Nowadays, if we visit a new place and we are not aware of the language then it is not a problem at all, as for this also machine learning helps us by converting the text into our known languages. Google's GNMT (Google Neural Machine Translation) provide this feature, which is a Neural Machine Learning that translates the text into our familiar language, and it called as automatic translation.
- The technology behind the automatic translation is a sequence to sequence learning algorithm, which is used with image recognition and translates the text from one language to another language.

Thank You